Izrada web aplikacije za upravljanje poslovima kozmetičkog salona u PHP i JavaScript razvojnim okruženjima

Štrucelj, Leona

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:298002

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-11-30**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet informatike

LEONA ŠTRUCELJ

IZRADA WEB APLIKACIJE ZA UPRAVLJANJE POSLOVIMA KOZMETIČKOG SALONA U PHP I JAVASCRIPT RAZVOJNIM OKRUŽENJIMA

Završni rad

Pula, rujan 2019. godine

Sveučilište Jurja Dobrile u Puli
Fakultet informatike

LEONA ŠTRUCELJ

IZRADA WEB APLIKACIJE ZA UPRAVLJANJE POSLOVIMA KOZMETIČKOG SALONA U PHP I JAVASCRIPT RAZVOJNIM OKRUŽENJIMA

Završni rad

JMBAG: 0303069568, redovna studentica
Studijski smjer: Sveučilišni preddiplomski studij Informatika
Predmet: Programsko inžinjerstvo
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija
Mentor: doc.dr.sc. Tihomir Orehovački

Pula, rujan 2019. godine

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana _____, kandidat za prvostupnika _____ ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____, _____ godine

SAŽETAK

Cilj ovog završnog rada je izraditi Web aplikaciju za upravljanje poslovima kozmetičkog salona. Ideja za Web aplikacijom ove vrste je potekla od bavljenja geliranja noktiju u slobodno vrijeme, a s obzirom na kompleksnost pri vođenju rezervacija, javljanja na pozive i slično, javila se potreba za smanjenje takvih obveza na način da korisnici naprave sami taj posao online pomoću navedene aplikacije.

U današnje vrijeme gotovo svi imaju pristup internetu i na taj način mogu pristupiti i koristiti Web aplikaciju. Upotrebom PHP tehnologije za pozadinski dio (eng. Backend), te HTML, CSS, Javascript i jQuery tehnologije za vidljivi dio (eng. Frontend) nastoji se omogućiti korisniku da rezervira svoj termin online, pošalju upit ukoliko ih nešto zanima, vide dosadašnje radove i dr. Kao HTTP poslužitelj odabrati ćemo Apache, a za pohranu koristiti će se MySQL baza podataka te Web Storm kao besplatan alat za uređivanje te oblikovanje projekta i PHP koda. Dizajn Web stranice raditi ćemo kombinacijom HTML koda i CSS stila. Sve slike pohranjene u galeriji biti će privatno vlasništvo, a obrađivati ćemo ih u programu za obradu slika. Za modeliranje i oblikovanje baze podataka koristi ćemo XAMPP web server otvorenog koda jer se sastoji od MySql baze podataka i Apache web servera, a ima mogućnost interpretacije skripti u PHP programskom jeziku. Instalacijom XAMPP-a automatski je dobiven i phpMyAdmin koji služi za jednostavno upravljanje bazom podataka.

Ključne riječi: PHP aplikacija, HTML, CSS, Javascript, Web Storm, MySQL database, upravljanje kozmetičkim salonom, online rezervacija

ABSTRACT

The goal of this final paper is to create a Web application which will manage the business of a beauty salon. The idea for this kind of a Web application originated from doing nail gel work in free time, and given the complexity of managing reservations, answering calls and so on, the need has arisen to decrease the amount of those responsibilities by enabling customers to do those things themselves online through the said application.

Nowadays, almost everyone has an access to the internet so they can access and use the Web application. Using PHP technology for Backend, as well as HTML, CSS, Javascript and jQuery technology for Frontend to enable customers to book an appointment online, send an inquiry if they have any questions, see previous pieces of work and so on. As the HTTP server we will select Apache, and for storage we will be using MySQL database as well as free tool Web Storm for editing and formatting the project and PHP code. The design of the Website will be made by the combination of HTML code and CSS style. All images stored in the gallery will be privately owned and will be processed through the program for image processing. We will use the open source XAMPP web server to model and format the database, as it consists of a MySql database and an Apache web server, and has the ability to interpret scripts in the PHP programming language. By installing XAMPP, phpMyAdmin is also automatically obtained and is used for an easy database management.

Key words: PHP application, HTML, CSS, Javascript, Web Storm, MySQL database, managing the beauty salon, online reservation

SADRŽAJ

1. UVOD 1
2. RAZRADA TEME 3
3. DIJAGRAMI 4
   3.1. Dijagram slučajeva korištenja (Use Case Diagram) 4
   3.2. ER model 6
4. OPIS FUNKCIONALNOSTI 8
   4.1. Početna stranica 8
   4.2. Prijava 15
   4.3. Registracija 18
   4.4. Kontakt 22
   4.5. Galerija 25
   4.6. Rezervacija 28
   4.7. Cjenik 34
   4.8. Odjava 35
5. BUDUĆI RAD NA WEB APLIKACIJI 36
6. ZAKLJUČAK 37
LITERATURA 39
POPIS SLIKA 40

1. UVOD

U posljednjih nekoliko godina dogodio se veliki napredak u razvoju usluga koje su dostupne svim korisnicima putem WWW-a[1].

U praksi se upravljanje poslovima kozmetičkog salona, točnije online rezervacije, još uvijek bilježe klasično papirom i olovkom. Taj pristup se pokazao da oduzima jako puno vremena te da se događa velik broj „ljudskih pogrešaka“ pa bi kao rješenje trebala poslužiti Web aplikacija koja bi vlasnicima sustava omogućavala brži uvid u sadržaj takve vrste.

U radu je prikazana izrada Web aplikacije za upravljanje poslovima kozmetičkog salona, ali jednako tako se slična aplikacija može primijeniti na bilo koju uslužnu djelatnost. Ideja za Web aplikacijom ove vrste je potekla od autorskog bavljenja uređenja noktiju u slobodno vrijeme. Iz iskustva se pokazalo problematičnim proces naručivanja klijenata za što je do sad korišten manualni oblik, odnosno isto se vršilo papirom i olovkom, a s obzirom na kompleksnost pri vođenju rezervacija, javljanja na pozive i sl. javila se potreba za smanjenje takvih obveza na način da korisnici naprave sami taj posao online pomoću navedene aplikacije.

S obzirom da u današnje vrijeme gotovo svi imaju pristup internetu i na taj način mogu pristupiti i koristiti Web aplikaciju. U radu će biti prikazan, po fazama, razvoj aplikacije u kojoj ćemo upotrebom PHP tehnologije za pozadinski dio (eng. Backend), te HTML, CSS, Javascript i jQuery tehnologije za vidljivi dio (eng. Frontend).

Na taj način će se nastojati omogućiti korisniku da rezervira svoj termin online, pošalje upit ukoliko ga nešto zanima, pogleda dosadašnje radove i dr. Kao HTTP poslužitelj biti će odabran Apache, a za pohranu koristiti će se MySQL baza podataka.

Također ćemo vidjeti kako je praktičan Web Storm kao besplatan alat za uređivanje te oblikovanje projekta i PHP koda. Dizajn Web stranice izraditi ćemo kombinacijom HTML koda i CSS stila.

Sve slike koje su korištene u aplikaciji i koje su pohranjene u galeriji su privatno vlasništvo, a obrađene su u programu za obradu slika. Nadalje, pri izradi aplikacije, za modeliranje i oblikovanje baze podataka korišten je XAMPP web server otvorenog

[1] engl. World Wide Web, skraćeno Web

koda. XAMPP server se sastoji od MySql baze podataka i Apache web servera, a ima mogućnost interpretacije skripti u PHP programskom jeziku.

Prilikom pokretanja XAMPP-a potrebno je uključenje Apache web servera i MySQL baze podataka kako bi Web aplikacija radila i kako bi njen sadržaj mogao biti promijenjen. Instalacijom XAMPP-a automatski je dobiven i phpMyAdmin koji služi za jednostavno upravljanje bazom podataka.

Cilj ovog završnog rada je izrada Web aplikaciju za upravljanje poslovima konkretnog kozmetičkog salona, vodeći računa o potrebama istoga, a koja bi vlasnicima salona kao i korisnicima usluge omogućila upravljanje rezervacijama te prikaz istih.

2. RAZRADA TEME

Aplikacija je napravljena od dva dijela. Prvi dio je namijenjen neprijavljenim korisnicima, točnije onima koji nisu izvršili registraciju, te ne zahtjeva prijavu u sustav. Svrha prvog dijela je prijava korisnika u sustav ukoliko već ima kreiran račun, a ako ga ne posjeduje može izraditi novi korisnički račun. Neprijavljeni korisnici imaju mogućnost kontaktirati vlasnika (administratora) pomoću forme ukoliko trebaju neke informacije oko usluga i slično.

Drugi je dio namijenjen svim registriranim korisnicima i zahtjeva prijavu u sustav. Na tom dijelu korisnik pregledava postojeće sadržaje, može vidjeti galeriju dosadašnjih radova što mu može uvelike pomoći pri donošenju odluke želi li rezervirati termin u navedenom salonu. Sve usluge definirane su u cjeniku usluga koji se nalazi u kartici „cjenik“. Korisniku je također na raspolaganju uvid u zauzete termine na temelju kojih može napraviti novu rezervaciju za neku od ponuđenih usluga. Pri rezervaciji novih termina otvara se novi okvir u kojem je korisnik dužan popuniti sva prazna polja. U praznim poljima se traži unos datuma i vremena kada korisnik želi rezervirati termin kao i lista svih usluga u padajućem izborniku. Korisnik upisuje i broj mobilnog uređaja putem kojeg ga vlasnik salona može kontaktirati u slučaju promjena i sl., a i ukoliko korisnik možda želi rezervirati termin za nekog drugog člana putem svog korisničkog profila.

Administrator ima najopsežniji uvid u aplikaciju, za sada ne toliko drukčiji od korisničkog, no nudi mu dodatno prikaz imena korisnika rezervacija jer zbog zaštite korisničkih podataka korisnici nemaju ovlasti vidjeti isti.

U radu ćemo opisati sve funkcionalnosti koje smo prethodno naveli. U nastavku ćemo započeti s dijagramom slučajeva korištenja te ER modelom u kojem se vidi prikaz povezanih tablica podataka. Slijedi prikaz aplikacije i koda kojeg ćemo također obraditi. Pred sami kraj navodimo buduća poboljšanja i nadogradnje aplikacije te dolazimo do zaključka gdje smo objedinili sve u jedno i na temelju toga donijeli zaključke o izrađenoj aplikaciji, njenoj svrsi kao i okruženjima u kojima je kreirana.

3. DIJAGRAMI

Dijagrami su vrlo vrijedan alat s kojim pojednostavljeno možemo grafički, odnosno slikovno prikazati ponašanje sustava ili dijelove sustava te na taj način jasnije vidjeti izazove u istima, a što će nam pomoći kod rješenja nekoga problema. Koriste se kako bi se pojasnila logika cjelokupne problematike te kako bi se bilo u mogućnosti ponuditi rješenje za postojeći problem.

U nastavku rada ćemo najprije, grafički prikazati dionike u procesu te kanale komunikacije. Također, iz dijagrama ćemo imati jasniju percepciju kako funkcionira poslovni subjekt čiju potrebu za aplikacijom želimo zadovoljiti.

Važno je reći da ovakav dijagram ne pojašnjava način rada, već samo daje jasniju sliku na potrebe koje korisnika ima.

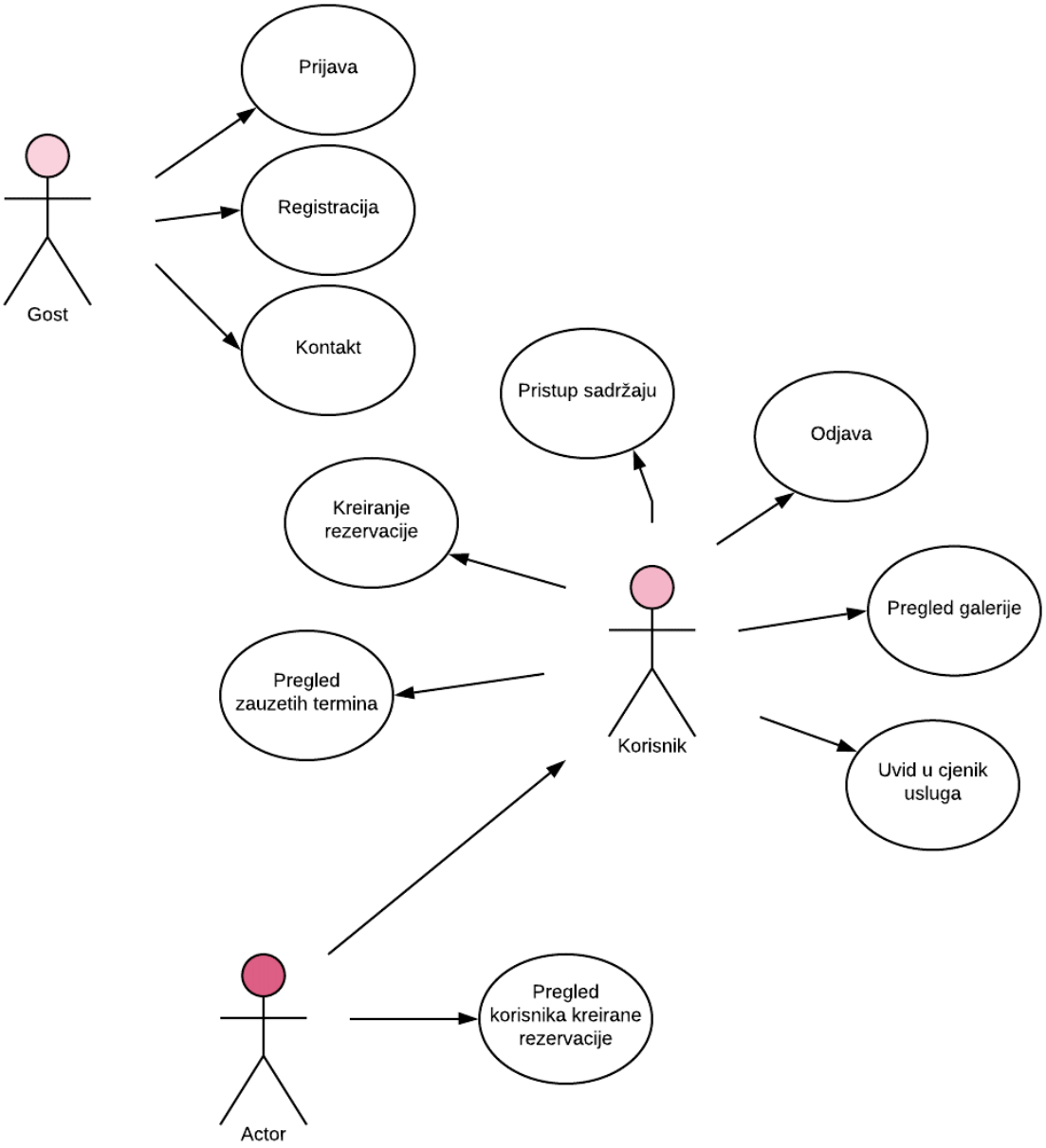
3.1. Dijagram slučajeva korištenja (Use Case Diagram)

U konkretnom slučaju, pri izradi aplikacije za kozmetički salon, slijede odnosi koje treba razumjeti, a koji su prikazani u dijagramu slučaja korištenja. Na slici 1. prikazan je dijagram slučajeva korištenja za sve tipove korisnika. Tako se iz dijagrama da vidjeti da imamo tri vrste korisnika koje smo nazvali: Gost, Korisnik i Actor.

Gost može vršiti sljedeće aktivnosti: Prijava, Registracija, Kontakt.

Korisnik, misli se pritom na one koji su već koristili uslugu i već su prijavljeni u aplikaciji, može provoditi sljedeće: Vršiti rezervaciju, Pregledati zauzete termine, Pristup sadržaju, Odjava, Pregled galerije, Uvid u cjenik usluga.

Actor pak ima uvid u cjelokupnu aplikaciju te može vidjeti potpuni pregled korisnika kreirane rezervacije.



Slika 1.: Use case diagram (Izvor: Autorski rad)

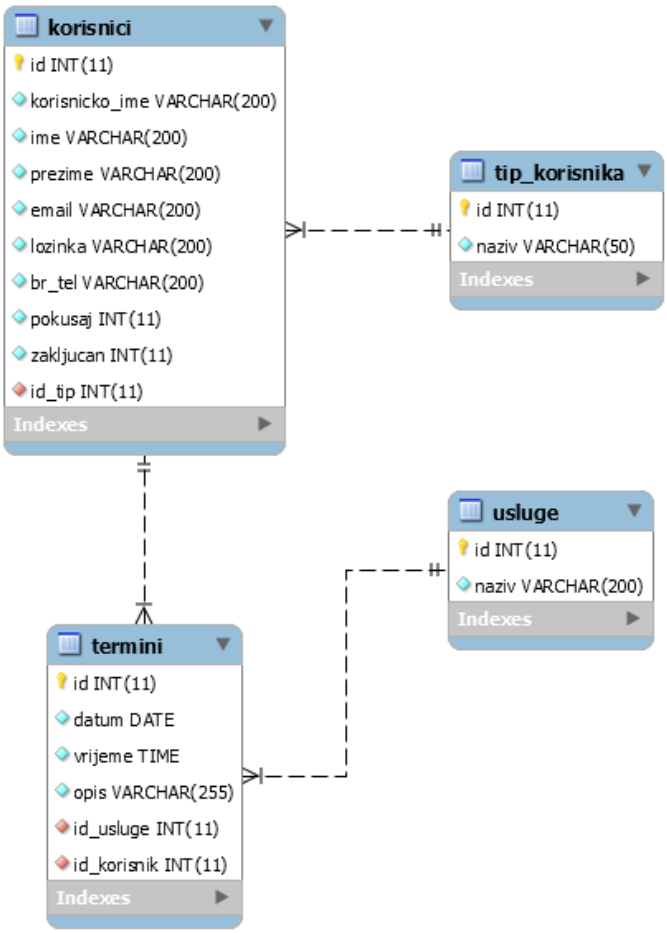
Slijedom prije navedenog, gost je neprijavljeni korisnik i kao takvome su mu prava u aplikaciji ograničena. Onemogućeno mu je pregledavanje i stavljanje bilo kakvog sadržaja. Jedino što mu je omogućeno je prijava ukoliko ima korisnički račun, izrada korisničkog računa te kontakt vlasnika. Prijavljeni korisnik ima veća prava u odnosu na gosta. Omogućeno mu je pregledavanje i dodavanje sadržaja, točnije kreiranje rezervacije kao i prikaz kreiranih dok administrator može još dodatno vidjeti koji je korisnik kreirao rezervaciju. Korisniku takva mogućnost nije odobrena zbog zaštite podataka drugih korisnika koji su kreirali svoju rezervaciju.

3.2. ER model

Baza podataka predstavlja logičku organizaciju podataka koja se odnosi na vrste podataka i njihove međusobne veze. ER-a model je prikaz znanja određenim samo-oblicima o objektima, vezama i svojstvima (Varga, 2010.). Uvidom u ER model može se vidjeti struktura baze podataka, relacije i atributi te veze između njih.

PhpMyAdmin korišten je za kontroliranje i administraciju MySQL-a i dobiven je automatski s instalacijom XAMPP-a. MySQL je alat napisan u PHP programskom jeziku. SQL je jezik koji omogućuje rad s bazama podataka, odnosno SQL je *lingua france* u svijetu baza podataka (Rabuzin, 2011.).

Slika 2. prikazuje ER model dok u nastavku slijedi kratki opis svake tablice.



Slika 2.: ER model (Izvor: Autorski rad)

Tablice u aplikaciji su međusobno povezane relacijama radi lakšeg i bržeg pretraživanja te očuvanja integriteta baze podataka. Svi podaci se spremaju na server kako bismo imali potpunu kontrolu nad njima.

Nadalje, tablica korisnik služi kao popis svih registriranih korisnika aplikacije i u nju se spremaju svi osobni podaci korisnika. Tablica korisnik je centralna tablica u bazi podataka jer se na nju vežu sve druge tablice. Primarni ključ tablice je „id“ koji je cjelobrojnog tipa i postavljen je na automatsko povećavanje (eng. Auto increment). Atribut „pokušaj“ služi za praćenje koliko se korisnik puta krivo prijavio, tj. koliko puta je pogriješio lozinku. Ukoliko korisnik 4. puta upiše krivo lozinku njegov profil će se zaključati, točnije biti će blokiran, i neće biti u mogućnosti prijaviti se s postojećim računom. To možemo iščitati i iz atributa „zaključan“ ukoliko je indeks postavljen na broj 1.

„id_tip“ je vanjski ključ na tablicu tip_korisnika koja u sebi sadrži „id“ kao primarni ključ tablice te „naziv“ po kojem možemo znati da li je korisnik admin ili nije. Ukoliko je korisnikov „id“ = 3 znači da je korisnik admin, a ukoliko je „id“ = 1 znači da je običan korisnik. Korisnik može biti točno jednog tipa, a tip korisnika može pripadati barem jednom ili više korisnika pa je veza između tih tablica M:1. Također, korisnik ima mogućnost rezervacije jednog ili više termina, a termin može pripadati točno jednom korisniku.

Tablica termini sadrži primarni ključ „id“ te dva vanjska ključa, a to su „id_korisnik“ koji je povezan na tablicu „korisnici“ te „id_usluge“ povezan na tablicu usluge. Svaki termin ima svoj datum i vrijeme, a preko vanjskog ključa možemo saznati za koju je točno uslugu korisnik rezervirao. Termin može pripadati točno jednoj usluzi, a svaka usluga može imati više termina pa je veza među tablicama M:1.

4. OPIS FUNKCIONALNOSTI

Izrađena aplikacija sastoji se od više stranica koje nude različite funkcionalnosti. Sljedeći odlomci opisuju mogućnosti pojedinih stranica.

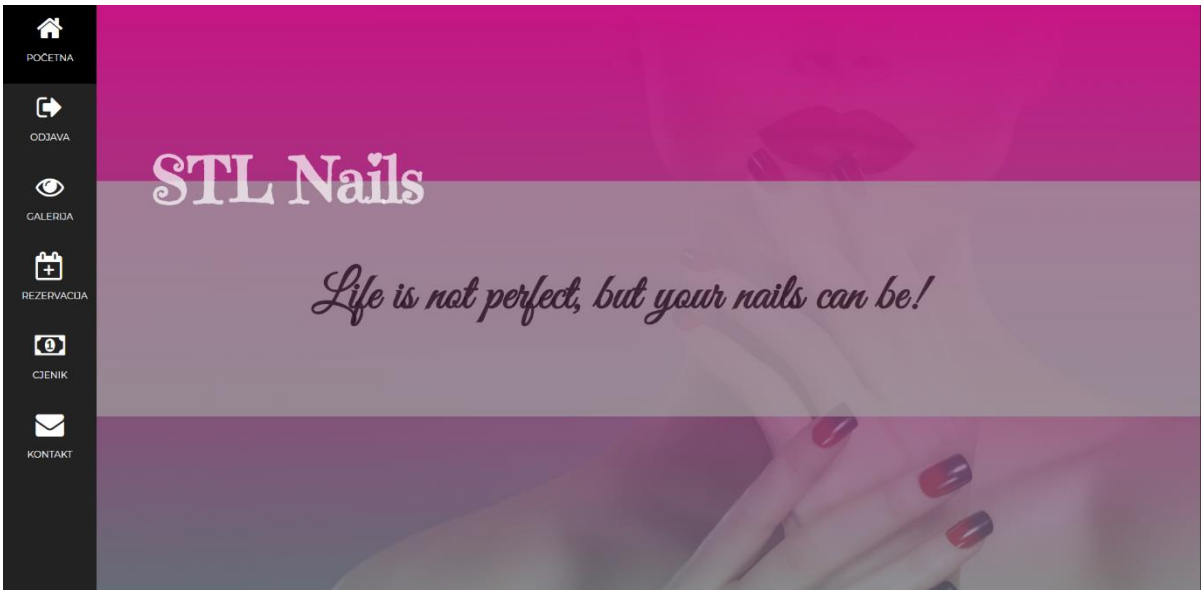
4.1. Početna stranica

Početna stranica neprijavljenim korisnicima (gostima) nudi mogućnost prijave, registracije te kontakta vlasnika salona. Za dizajn navigacijske trake koristili smo komponente iz W3.css okvira (eng. Framework) koji je uvelike ubrzao i pojednostavio razvoj web aplikacije. Font slova promijenili smo pomoći „Google Fonts-a“ čiji link uključimo u „header“ datoteke. Za svaku od bočnih kartica u navigacijskoj traci provjerava se postoji li prijavljen korisnik kako bi se ovisno o tome prikazale ili sakrile pojedine kartice. Npr. kartica odjave će ostati sakrivena sve dok je u funkciji „dajKorisnika()“ unutar klase „Sesija“ vrijednost jednaka nuli.

Slika 3. prikazuje izgled početne stranice prije izvršene prijave,

Slika 3.: Početna stranica prije izvršene prijave (Izvor: Autorski rad)

dok slika 4. nudi prikaz početne stranice nakon što se korisnik prijavi.



Slika 4.: Prikaz početne stranice nakon prijave (Izvor: Autorski rad)

Prijavljenim korisnicima se u bočnom izborniku nudi mogućnost odabira između nekoliko kartica prikazanih na slici 4. To su kartice početne stranice, gumba odjave, galerije fotografija, prikaza rezervacija i obrasca za rezervaciju termina, cjenika usluga i forme za kontakt vlasnika salona.

Slike 5. i 6. prikazuju kod napisan za dizajn i funkcionalnost početne stranice.

pocetna.php

```
<body class="w3-black" id="popravi">

<div class="w3-display-bottomleft w3-container w3-amber w3-gray w3-hide-small"
   style="bottom:30%;opacity:0.7;width:100%;margin-left:120px;height:40%">
  <h2 id="tekstpocetna"><b>Life is not perfect, but your nails can be!</b></h2>
</div>

<div class="w3-display-bottomleft w3-hide-small"
   style="bottom:70%;opacity:0.7;width:100%;margin-left:120px;height:10%">
  <h2 id="logo"><b>STL Nails</b></h2>
</div>

<!-- Icon Bar (Sidebar - hidden on small screens) -->
<nav class="w3-sidebar w3-bar-block w3-small w3-hide-small w3-center">

  <a href="pocetna.php" class="w3-bar-item w3-button w3-padding-large w3-black">
    <i class="fa fa-home w3-xxlarge"></i>
    <p>POČETNA</p>
  </a>

  <?php
  if(Sesija::dajKorisnika() == null ){
  echo "<a href=\"prijava.php\" class=\"w3-bar-item w3-button w3-padding-large w3-hover-black\" id=\"prijava_gumb\">
    <i class=\"fa fa-user w3-xxlarge\"></i>
    <p>PRIJAVA</p>
    </a>
    <a href=\"registracija.php\" class=\"w3-bar-item w3-button w3-padding-large w3-hover-black\">
       <i class=\"fa fa-arrow-circle-right w3-xxlarge\"></i>
       <p>REGISTRACIJA</p>
     </a>";
    }
    else {
    echo "<a href=\"odjava.php\" class=\"w3-bar-item w3-button w3-padding-large w3-hover-black\" id=\"odjava_gumb\">
            <i class=\"fa fa-sign-out w3-xxlarge\"></i>
```

Slika 5.: Prikaz koda početne stranice_prvi dio (Izvor: Autorski rad)

pocetna.php

```
<body class="w3-black" id="popravi">

<div class="w3-display-bottomleft w3-container w3-amber w3-gray w3-hide-small"
   style="bottom:30%;opacity:0.7;width:100%;margin-left:120px;height:40%">
  <h2 id="tekstpocetna"><b>Life is not perfect, but your nails can be!</b></h2>
</div>

<div class="w3-display-bottomleft w3-hide-small"
   style="bottom:70%;opacity:0.7;width:100%;margin-left:120px;height:10%">
  <h2 id="logo"><b>STL Nails</b></h2>
</div>

<!-- Icon Bar (Sidebar - hidden on small screens) -->
<nav class="w3-sidebar w3-bar-block w3-small w3-hide-small w3-center">

  <a href="pocetna.php" class="w3-bar-item w3-button w3-padding-large w3-black">
    <i class="fa fa-home w3-xxlarge"></i>
    <p>POČETNA</p>
  </a>

  <?php
  if(Sesija::dajKorisnika() == null ){
  echo "<a href=\"prijava.php\" class=\"w3-bar-item w3-button w3-padding-large w3-hover-black\" id=\"prijava_gumb\">
    <i class=\"fa fa-user w3-xxlarge\"></i>
    <p>PRIJAVA</p>
    </a>
    <a href=\"registracija.php\" class=\"w3-bar-item w3-button w3-padding-large w3-hover-black\">
       <i class=\"fa fa-arrow-circle-right w3-xxlarge\"></i>
       <p>REGISTRACIJA</p>
     </a>";
    }
    else {
    echo "<a href=\"odjava.php\" class=\"w3-bar-item w3-button w3-padding-large w3-hover-black\" id=\"odjava_gumb\">
              <i class=\"fa fa-sign-out w3-xxlarge\"></i>
            <p>ODJAVA</p>
          </a>"
          ;
 }
 ?>

 <?php
   if(Sesija::dajKorisnika() != null ){
    echo "<a href=\"galerija.php\" class=\"w3-bar-item w3-button w3-padding-large w3-hover-black\" id=\"odjava_gumb\">
                <i class=\"fa fa-eye w3-xxlarge\"></i>
                <p>GALERIJA</p>
            </a>

            <a href=\"rezervacija.php\" class=\"w3-bar-item w3-button w3-padding-large w3-hover-black\" id=\"odjava_gumb\">
                 <i class=\"fa fa-calendar-plus-o w3-xxlarge\"></i>
               <p>REZERVACIJA</p>
               </a>
             <a href=\"tablica.php\" class=\"w3-bar-item w3-button w3-padding-large w3-hover-black\" id=\"odjava_gumb\">
               <i class=\"fa fa-money w3-xxlarge\"></i>
               <p>CJENIK</p>
             </a>";
  }
  ?>

 <a href="upit.php" class="w3-bar-item w3-button w3-padding-large w3-hover-black">
   <i class="fa fa-envelope w3-xxlarge"></i>
   <p>KONTAKT</p>
  </a>
</nav>

</body>
</html>
```

Slika 6.: Prikaz koda početne stranice_drugi dio (Autorski rad)

Nakon što smo dodali pozadinsku sliku u .css dokumetu, definiranu unutar zagrada id-a „popravi“, za privlačniji izgled korisnicima uključili smo linearni gradijent kombinacije ružičaste i crne boje. Od 47 do 50 linije koda stvorili smo „div“ za početni tekst na stranici. Također, isto smo napravili i za logo salona kojeg smo u stilu pomicali sve dok ne bi došao u gornji desni kut prozora. Sve klase tagova nalaze se u .css dokumentima koji su uključeni na samom vrhu stranice kao i pripadajući id u kojima je definiran dizajn koji će navedeni element poprimiti.

Unutar php tagova provjerava se da li je korisnik prijavljen, a ukoliko ga nema neće mu biti prikazani navedene kartice u bočnoj traci. Za provjeru smo morali uključiti datoteku „sesija.class.php“ čije funkcije i pozivamo za samu provjeru. U 66 liniji koda provjeravamo da li je sesija danog korisnika == 0, ukoliko je znači da ne postoji prijavljeni korisnik i tada će mu biti prikazane kartice za prijavu i registraciju, a u protivnom ako je prijavljen stajati će mu gumb odjave. Klasa „Sesija“, u kojoj se nalaze sve funkcije koje pozivamo u datoteci početne stranice, prikazujemo na slici 7. u nastavku.

Slika 7.: Prikaz funkcija unutar sesija.class.php (Izvor: Autorski rad)

Klasa „Sesija“ sadrži statičku funkciju „dajKorisnika()“ u kojoj se provjerava da li je korisnik postavljen, ukoliko je sprema ga u varijablu „korisnik“ te istu vraća u „returnu“ funkcije. Konstantu korisnika definirali smo na početku php dokumenta. Uz funkciju „dajKorisnika()“ kreirali smo i druge funkcije poput „kreirajSesiju()“ za samo kreiranje sesije ukoliko je prazna, tj. ukoliko ne postoji, a ista započinje pozivom metode „session_start()“ u liniji 14. Tu su i ostale funkcije koje koristimo tokom cijelog razvoja aplikacije. Sa slike 8. možemo vidjeti uključivanje datoteke iz koje se poziva funkcija kreirajKorisnika() unutar klase „Sesija“.

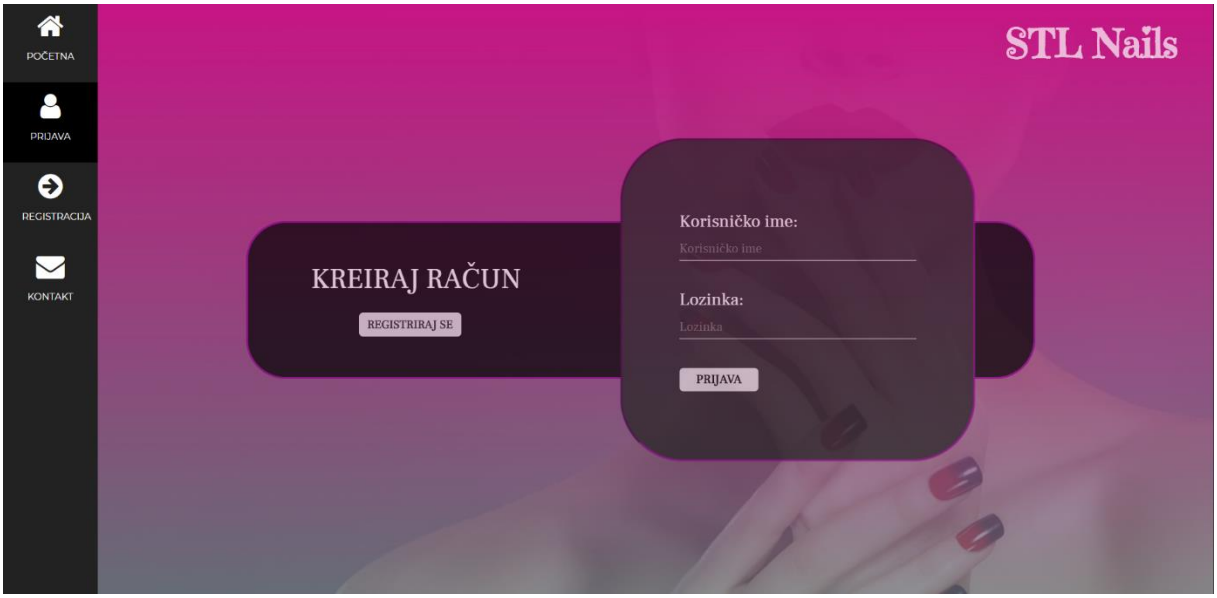
```php
<!DOCTYPE html>
<?php
include("baza.class.php");
include("sesija.class.php");
Sesija::kreirajSesiju();
$baza = new Baza();
$baza->spojiDB();

?>
```

Slika 8.: Poziv dokumenta sesija.class.php u drugi dokument (Izvor: Autorski rad)

Prvo smo uključili datoteku „baza.class.php“ u kojoj nam se nalaze funkcije za otvaranje, zatvaranje, ažuriranje baze i sl. jer bez njih ne bismo mogli ni dohvatiti podatke o korisniku. Slijedi uključenje dokumenta „sesija.class.php“ koju smo objasili i prije i bez koje ne bismo mogli koristiti funkcije za provjeru postoji li prijavljen korisnik.

4.2. Prijava

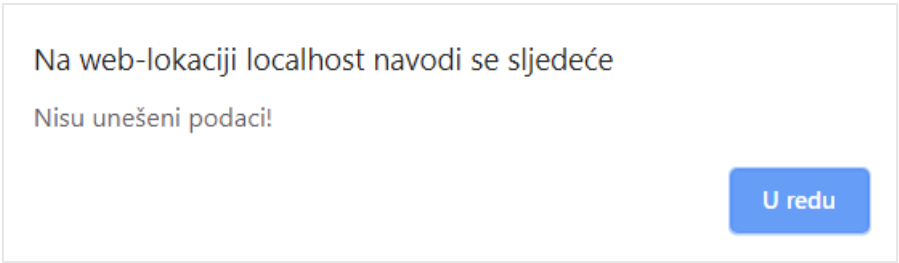
Gost se prijavljuje upisujući svoju Email adresu i lozinku u formularu na stranici prijave koja je prikazana na slici 9.



Slika 9.: Forma prijave (Izvor: Autorski rad)

S obzirom da se radi o gostu koji možda ne posjeduje korisnički račun, ponuđena mu je i opcija registracije koja se nalazi uz samu formu prijave te klikom na gumb „registriraj se“ program će ga automatski preusmjeriti na karticu registracije koju ćemo prikazati i objasniti u nastavku ovoga rada.

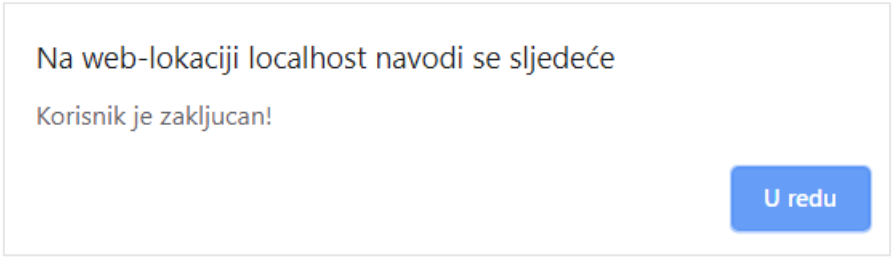
Ukoliko je gost ostavio neko od polja prazno u formi prijave ispisuje mu se poruka da nisu uneseni podaci što vidimo na slici 10.



Slika 10.: Prikaz poruke praznih polja (Izvor: Autorski rad)

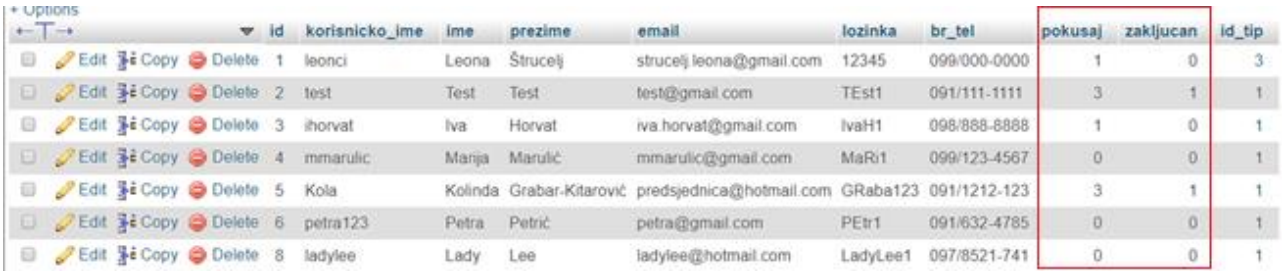
Ako gost pogriješi e-mail ili lozinku izbaciti će mu se poruka da korisnik nije registriran.

Također, kao što smo naveli na samom vrhu ovoga rada, kada korisnik 4. puta unese krivu lozinku njegov korisnički račun će se blokirati i neće biti više u mogućnosti prijaviti se s tim korisničkim računom. Ovo ograničenje smo napravili radi sigurnosti korisnika jer su u današnje vrijeme sve češće pojave hakiranja računa, a kada bismo u budućnosti implementirali kartično plaćanje usluge online to nikako ne bi bilo dobro za stvarnog korisnika računa.



Slika 11.: Prikaz poruke blokiranog korisnika (Izvor: Autorski rad)

To možemo vidjeti i u bazi sa slike 12. gdje je broj pokušaja postavljeno na 3, a zaključan na 1 što bi značilo da su takvi korisnici blokirani. Naravno, korisnika se može naknadno otključati ukoliko vlasnik salona promjeni podatak u retku iz broja 3 u neki manji broj pod stupcem „pokusaj“ te ako u stupcu „zakljucan“ prostavi vrijednost ponovno na nulu.



+ Options

	id	korisnicko_ime	ime	prezime	email	lozinka	br_tel	pokusaj	zakljucan	id_tip
Edit Copy Delete	1	leonci	Leona	Štrucelj	strucelj.leona@gmail.com	12345	099/000-0000	1	0	3
Edit Copy Delete	2	test	Test	Test	test@gmail.com	TEst1	091/111-1111	3	1	1
Edit Copy Delete	3	ihorvat	Iva	Horvat	iva.horvat@gmail.com	IvaH1	098/888-8888	1	0	1
Edit Copy Delete	4	mmarulic	Manja	Marulić	mmarulic@gmail.com	MaRi1	099/123-4567	0	0	1
Edit Copy Delete	5	Kola	Kolinda	Grabar-Kitarović	predsjednica@hotmail.com	GRaba123	091/1212-123	3	1	1
Edit Copy Delete	6	petra123	Petra	Petrić	petra@gmail.com	PEtr1	091/632-4785	0	0	1
Edit Copy Delete	8	ladylee	Lady	Lee	ladylee@hotmail.com	LadyLee1	097/8521-741	0	0	1

Slika 12.: Prikaz blokiranih korisnika u bazi (Izvor: Autorski rad)

Na slikama 13. i 14. možemo vidjeti kod u kojem smo definirali sva ograničenja pri prijavi u aplikaciju.

```php
if(Sesija::dajKorisnika() != null ){
    header("Location: http://localhost/zavrsni_rad/pocetna.php");
    die();
}

if(!empty($_POST)){
    $korime = $_POST['korisnicko_ime'];
    $p_lozinka = $_POST['prijava_lozinka'];

    if(empty($korime) || empty($p_lozinka)){
          $message = "Nisu uneseni podaci!";
          echo "<script type='text/javascript'>alert('$message');</script>";
    }
    else{
    $rez = $baza->selectDB('SELECT * FROM korisnici WHERE korisnicko_ime = "' .$korime. '" AND lozinka = "' .$p_lozinka. '";');
    if($rez->num_rows > 0){
            $rez = $rez->fetch_array();
            if($rez["zakljucan"]==1){
                $message = "Korisnik je zaključan!";
                echo "<script type='text/javascript'>alert('$message');</script>";
            }else{
                $rez = $baza->selectDB('SELECT * FROM korisnici WHERE korisnicko_ime = "' .$korime. '" AND lozinka = "' .$p_lozinka. '";');
                $korisnik=$rez->fetch_assoc();
                Sesija::kreirajKorisnika($korisnik);
                $baza->updateDB("UPDATE korisnici SET pokusaj=0 WHERE korisnicko_ime='{$korisnik["korisnicko_ime"]}'");
                header("Location: http://localhost/zavrsni_rad/pocetna.php");
                die();
            }
    }else{
            $rez = $baza->selectDB('SELECT * FROM korisnici WHERE korisnicko_ime = "' .$korime. '";');
            if($rez->num_rows > 0){
                $rez = $rez->fetch_array();
                $broj_p = intval($rez["pokusaj"]);
```

Slika 13.: Prikaz provjera i ograničenja uz prijavu (Izvor: Autorski rad)

```php
                if($broj_p >= 3){
                    $baza->updateDB("UPDATE korisnici SET zakljucan=1 WHERE korisnicko_ime='{$korime}'");
                    $message = "Korisnik je zaključan!";
                    echo "<script type='text/javascript'>alert('$message');</script>";
                }
                else{
                    $broj_p=$broj_p+1;
                    $baza->updateDB("UPDATE korisnici SET pokusaj='{$broj_p}' WHERE korisnicko_ime='{$korime}'");
                }
            }
            else {
                $message = "Korisnik nije registriran!";
                echo "<script type='text/javascript'>alert('$message');</script>";
            }
     }
     }
}
```

Slika 14.: Prikaz provjera i ograničenja uz prijavu_drugi dio (Izvor: Autorski rad)

Na samom početku koda, u 14. liniji imamo selekciju u kojoj ukoliko nije prazno polje postavljamo korisničko ime i lozinku pod „korisničko_ime“ i „prijava_lozinka“ pomoću varijable „_POST“. Varijabla $ _POST koristi se za prikupljanje vrijednosti iz obrasca metodom = "post". Informacije poslane iz obrasca metodom POST nevidljive su drugima i nemaju ograničenja u količini informacija koje treba poslati (3).

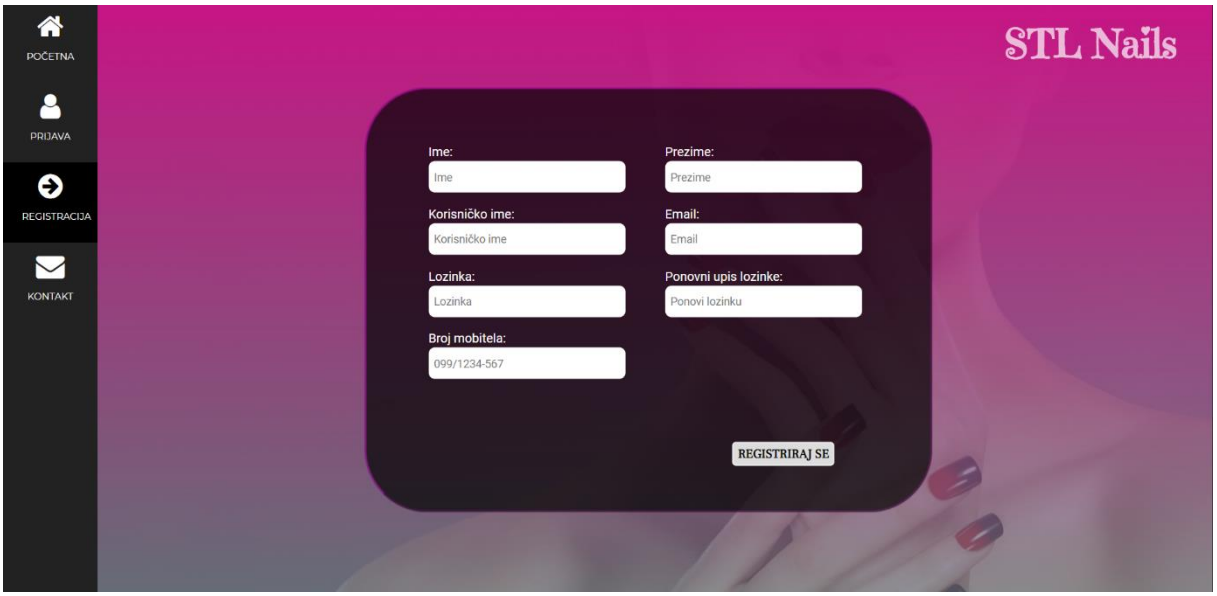
U 18. liniji ispitujemo da li je polje e-maila ili polje lozinke prazno, ukoliko je program će nam vratiti poruku da nisu uneseni svi podaci. Poruka se stvara pomoću „alert-a“ kojem smo predali svoju varijablu „$message“ u kojoj je i spremljen tekst poruke koja će se nalaziti u okviru.

Ukoliko su oba polja točno uspješno popunjena provjeravamo da li je korisnik blokiran, točnije da li je vrijednost u bazi pod „zaključan“ jednaka 1. Ukoliko korisnik nije blokiran i uspješno se prijavio postavljamo njegove vrijednosti pokušaja na 0. Dakle, ako je 2 puta krivo upisao lozinku, u bazi će biti pohranjena vrijednost 2, no ako 3. put dobro upiše lozinku, podatak u bazi će se ažurirati na vrijednost 0 i korisnik će ponovno moći pogriješiti lozinku 3 puta.

Korisnika automatski preusmjeravamo na početnu stranicu nakon uspješne prijave. U nastavku koda vršimo slične provjere vezane uz blokiranje korisnika nakon više od 3 kriva pokušava.

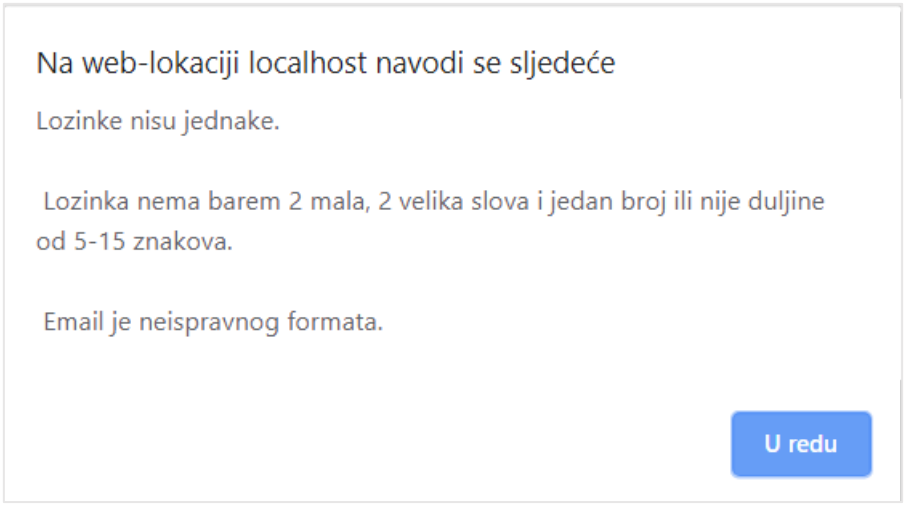
4.3. Registracija

Gost se registrira upisujući svoje podatke u formular za registraciju na stranici registracije koja je prikazana na slici 15. Sva polja su obavezna, a ukoliko gost neko od polja ostavi prazno izbaciti će mu se greška da nisu uneseni svi podaci kao i kod prijave.



Slika 15. Prikaz forme registracije (Izvor: Autorski rad)

Ukoliko gost upiše neko od polja krivo također će dobiti poruku greške koja je prikazana u nastavku na slici 16..



Slika 16.: Prikaz poruke krivo unesenih podataka u formi registracije (Izvor: Autorski rad)

Tako primjerice imamo provjeru ukoliko nisu jedne lozinke ili ukoliko nisu napisane po propisanim pravilima, ukoliko je e-mail neispravnog formata i sl. Svaku od tih poruka opisati ćemo u nastavku na kodu prikazanom na slikama 17. i 18.

```
11  if(isset($_POST['registracija'])){
12      if(!empty($_POST)){
13          $poruka = "";
14          //var_dump($_POST);
15
16          if(!isset($_POST['ime']) || !isset($_POST['prezime']) || !isset($_POST['korIme']) || !isset($_POST['email']) || !isset($_POST['lozinka'
17              $poruka .= "Nisu uneseni svi podaci. ";
18          }
19
20          else{
21              $ime = $_POST['ime'];
22              $prezime = $_POST['prezime'];
23              $korisnicko_ime = $_POST['korIme'];
24              $email = $_POST['email'];
25              $lozinka = $_POST['lozinka'];
26              $potvrda = $_POST['lozinka_provjera'];
27              $telefon = $_POST['brojMob'];
28              $id_tip = 1;
29
30              if(empty($ime) || empty($prezime) || empty($korisnicko_ime) || empty($email) || empty($lozinka) || empty($potvrda)){
31                              $poruka .= "Nisu uneseni svi podaci. ";
32                          }
33              else {
34                  $upit="SELECT * FROM korisnici WHERE korisnicko_ime = '{$korisnicko_ime}'";
35                  $res = $baza->selectDB($upit);
36
37                  if(strcmp($lozinka, $potvrda) != 0){
38                      $poruka .= "Lozinke nisu jednake.\\n\\n ";
39                  }
40                  $regex_lozinka = '/^(?=(?:.*[A-Z]){2,})(?=(?:.*[a-z]){2,})(?=(?:.*[0-9]){1,})\S{5,15}$/';
41                  if(!preg_match($regex_lozinka, $lozinka)){
42                      $poruka .= "Lozinka nema barem 2 mala, 2 velika slova i jedan broj ili nije duljine od 5-15 znakova.\\n\\n ";
43                  }
```

Slika 17.: Prikaz koda za formu registracije (Izvor: Autorski rad)

Dakle, iz koda možemo vidjeti provjeru u 16. liniji ukoliko je neko od polja prazno što provjeravamo s funkcijom „isset“, točnije negacijom iste, program ispisuje tekst da nisu uneseni svi podaci koji je spremljen u varijablu „poruka“ te se ista poziva na kraju svih selekcija u 87. liniji koda.

Ukoliko su sva polja popunjena, vršimo daljnje provjere koje smo spremili u varijable na vrhu dokumenta i koje međusobno uspoređujemo. Za provjeru ispravne lozinke i e-mail adrese koristimo „regex“ koji možemo sami kreirati i provjeriti na stranici regexr.com .

U 37. liniji koda pomoću funkcije „strcmp()“ uspoređujemo dvije lozinke. Ta nam funkcija služi za usporedbu dva niza. Funkcija je slična funkciji „strncmp()“, s time da kod nje možemo odrediti broj znakova iz svakog niza koji će se koristiti u usporedbi sa strncmp() (w3schools.com). Dakle, dobro je jer funkcija prepoznaje velika i mala slova što nam je ovdje iznimno potrebno. U 40. liniji koda slijedi regex oblika lozinke. Lozinka treba sadržavati barem 2 mala i 2 velika slova te jedan broj jer u suprotnom ćemo dobiti poruku greške. U 45. liniji koda provjeravamo da li je korisničko ime zauzeto. Korisničko ime smo spremili u varijablu „$res“ u 35. liniji koda. Ukoliko je u retku broj

1, znači da korisničko ime u bazi već postoji. Isto provjeravamo i za e-mail adresu u 51. liniji koda. Nadalje, u 56. liniji koda pomoću regex oblika provjeravamo da li je e-mail adresa ispravnoga formata. Provjeru vršimo pomoću negacije funkcije „preg_match()“ koja služi za podudaranje izraza, u našem slučaju podudaranje varijable „$mail“ u kojoj je spremljen regex oblik u liniji 56. s varijablom „$email“ u kojoj je spremljena upisana e-mail adresa u obrascu. Ukoliko se ta dva argumenta ne podudaraju program će izbaciti poruku greške. U nastavku je gotovo sve slično, u svakoj od selekcija koristimo funkciju „preg_match()“ s kojom uspoređujemo podudaranje dvije varijable.

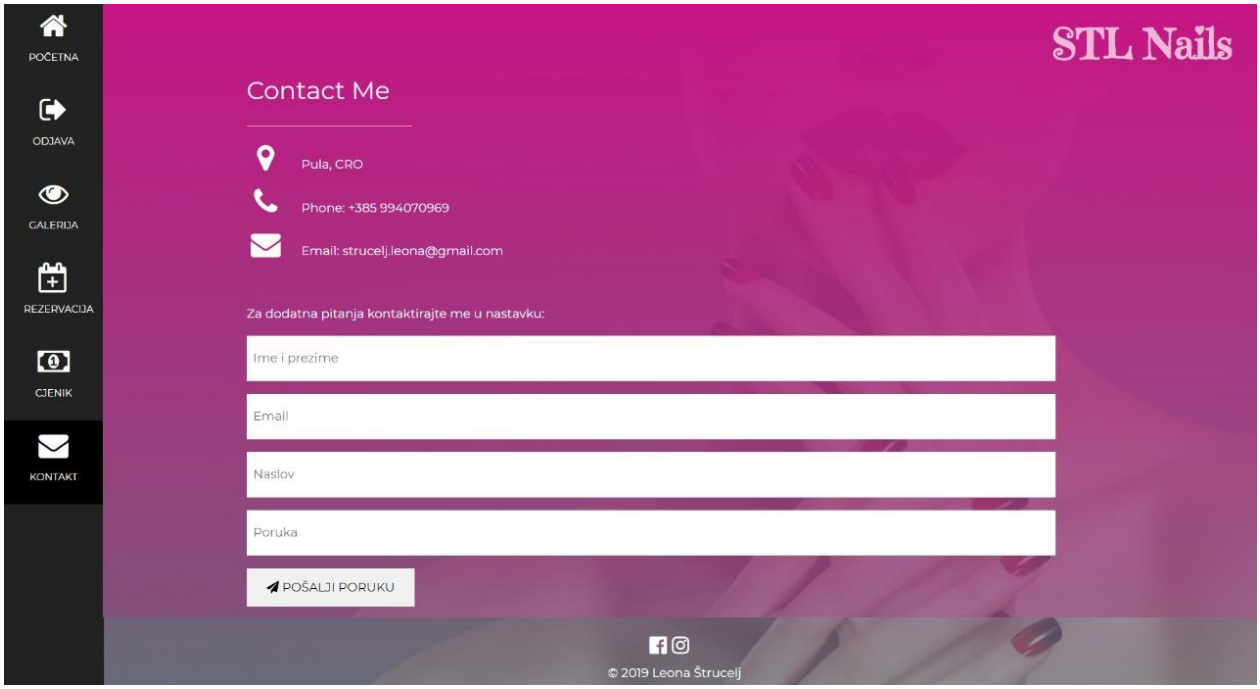
Na slici 18. možemo vidjeti ostatak koda provjere podataka upisanih u formu registracije.

```
43          }
44          if ($res->num_rows == 1) {
45              $poruka .= "Korisnicko ime je zauzeto.\\n ";
46          }
47
48          $upit="SELECT * FROM korisnici WHERE email = '{$email}' LIMIT 1";
49          $res2 = $baza->selectDB($upit);
50          if ($res2 != null){
51          if ($res2->num_rows == 1) {
52              $poruka .= "Email je zauzet!\\n ";
53          }
54          }
55
56          $mail = '/^\b[a-z0-9._%-]+@[a-z0-9.-]+\.[a-z]{2,4}\b$/i';
57          if(!preg_match($mail, $email)){
58              $poruka .= "Email je neispravnog formata.\\n ";
59          }
60
61          $regex_znak = '/[\\()\'{}!#"\/]/';
62          if(preg_match($regex_znak, $ime)){
63              $poruka .= "Ime ne smije sadrzavati nedozvoljene znakove!\\n ";
64          }
65          if(preg_match($regex_znak, $prezime)){
66              $poruka .= "Prezime ne smije sadrzavati nedozvoljene znakove!\\n ";
67          }
68          if(preg_match($regex_znak, $korisnicko_ime)){
69              $poruka .= "Korisnicko ime ne smije sadrzavati nedozvoljene znakove!\\n ";
70          }
71          if(preg_match($regex_znak, $email)){
72              $poruka .= "Email ne smije sadrzavati nedozvoljene znakove!\\n ";
73          }
74          if(preg_match($regex_znak, $lozinka)){
75              $poruka .= "Lozinka ne smije sadrzavati nedozvoljene znakove!\\n ";
```

Slika 18.:Prikaz koda za formu registracije_drugi dio (Izvor: Autorski rad)

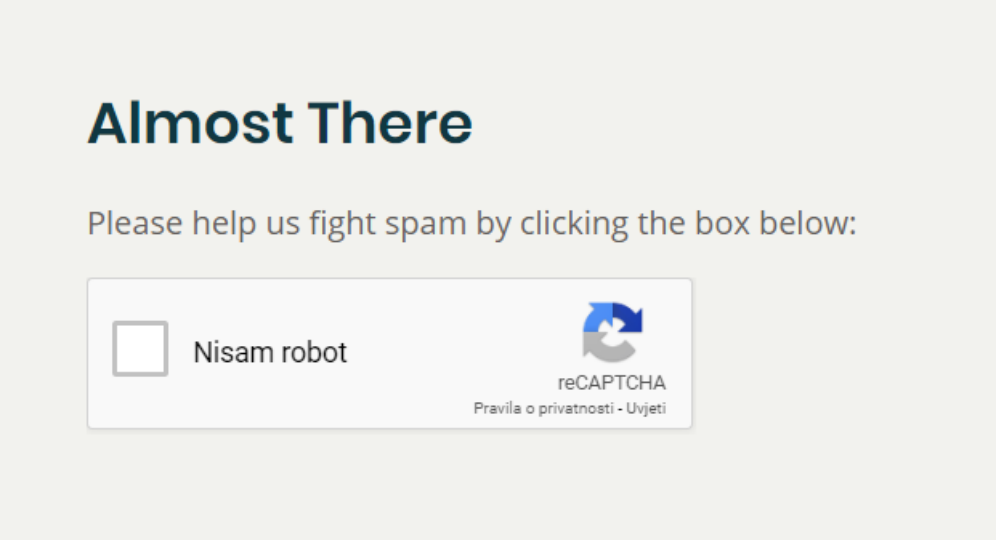
4.4. Kontakt

Kontakt formular prikazan na slici 19. je izrađen kako bi gosti i prijavljeni korisnici mogli kontaktirati vlasnika ukoliko imaju dodatnih pitanja ili ukoliko žele neke promjene. Formular se sastoji od polja za ime i prezime, e-mail adrese s koje će se poruka poslati, naslova poruke te same poruke. Poruka će stići na e-mail adresu vlasnika, ovdje ujedno i administratora, koja je naznačena na vrhu stranice za kontakt.



Slika 19.: Prikaz kontakt forme (Izvor: Autorski rad)

Za takvu komunikaciju koristili smo „Formspree“, internetska usluga s kojom spajamo naš obrazac za lakšu komunikaciju do E-mail adrese. Nakon klika na gumb za slanje poruke, korisniku se prikazuje forma potvrde gdje se provjerava da nije robot te ga se potvrdom preusmjerava ponovno na stranicu kontakta. Forma provjere je prikazana na slici 20.

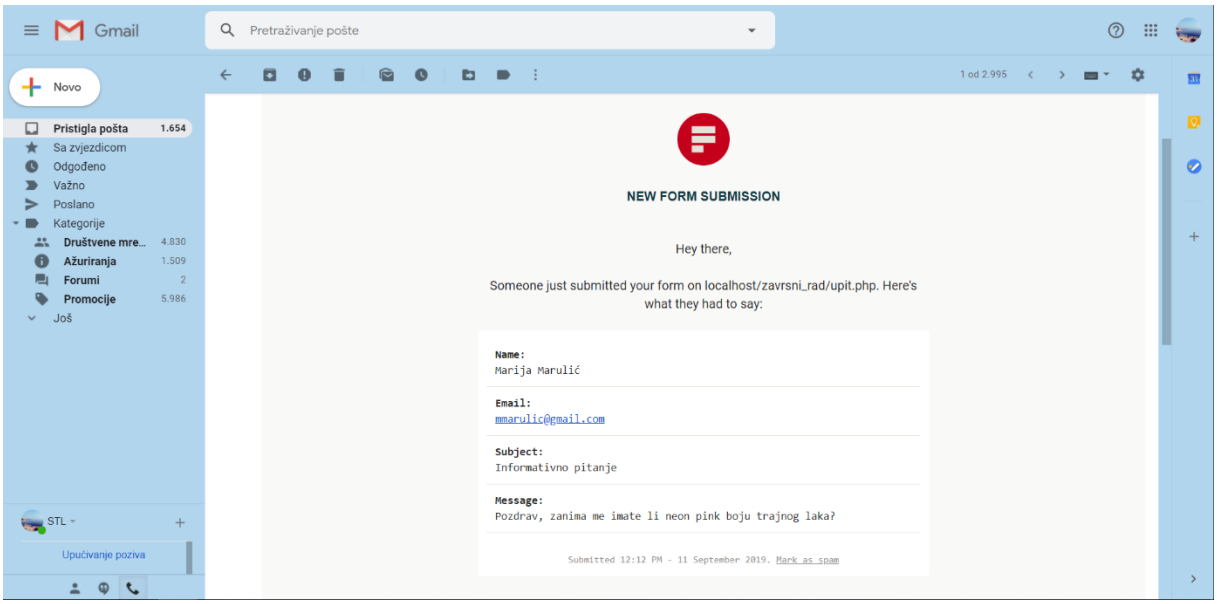


Slika 20.: Prikaz forme „Nisam robot“ (Izvor: Autorski rad)

Ukoliko je korisnik krivo upisao neki od unosa dobiti će poruku greške. Primjerice za upis krive e-mail adrese program će ga zatražiti ponovni upis. Takav prikaz dan nam je na slici 21.

Slika 21.: Prikaz poruke netočnog unosa e-mail adrese u kontakt formi (Izvor: Autorski rad)

Prikazana poruka stigla je na E-mail adresu i možemo je vidjeti na slici 22.



Slika 22.: Prikaz poruke na e-mail adresi vlasnika salona (Izvor: Autorski rad)

Kao što smo spomenuli, poruka se šalje preko formspree linka na e-mail adresu koju smo naveli u produžetku linka što možemo vidjeti i u 137 liniji koda na slici 22. Forma je dizajnirala pomoću komponenti iz W3.css stila.

Korisnici mogu kontaktirati vlasnika i putem Instagram ili Facebook društvene mreže pritiskom na ikonu u podnožju stranice. To možemo i iščitati iz 165 i 166 linije koda na slici 23. gdje se pritiskom na ikonu direktno preusmjerava na stranicu salona.

```
<!-- Contact Section -->
<form method="post" action="https://formspree.io/strucelj.leona@gmail.com">
<div class="w3-padding-64 w3-content w3-text-white" id="contact" style:"margin-right:50px">
  <h2 class="w3-text-light-grey">Contact Me</h2>
  <hr style="width:200px" class="w3-opacity">

  <div class="w3-section">
    <p><i class="fa fa-map-marker fa-fw w3-text-white w3-xxlarge w3-margin-right"></i> Pula, CRO</p>
    <p><i class="fa fa-phone fa-fw w3-text-white w3-xxlarge w3-margin-right"></i> Phone: +385 994070969</p>
    <p><i class="fa fa-envelope fa-fw w3-text-white w3-xxlarge w3-margin-right"> </i> Email: strucelj.leona@gmail.com</p>
  </div><br>
  <p>Za dodatna pitanja kontaktirajte me u nastavku: </p>

  <form target="_blank" style="margin-bottom:-9%">
    <p><input class="w3-input w3-padding-16" type="text" placeholder="Ime i prezime" required name="Name"></p>
    <p><input class="w3-input w3-padding-16" type="text" placeholder="Email" required name="Email"></p>
    <p><input class="w3-input w3-padding-16" type="text" placeholder="Naslov" required name="Subject"></p>
    <p><input class="w3-input w3-padding-16" type="text" placeholder="Poruka" required name="Message"></p>
    <p>
      <button class="w3-button w3-light-grey w3-padding-large" type="submit">
        <i class="fa fa-paper-plane"></i> POŠALJI PORUKU
      </button>
    </p>
  </form>
<!-- End Contact Section -->
</div>

  <!-- Footer -->
<footer class="w3-text-white w3-xlarge" id="podnozje">
  <a href="https://www.facebook.com/stl.leona.nails/"><i class="fa fa-facebook-official w3-hover-opacity" id="facebook"></i></a>
  <a href="https://www.instagram.com/stl.nails/"><i class="fa fa-instagram w3-hover-opacity"></i></a>
  <p class="w3-medium" id="ikone">© 2019 Leona Štrucelj </p>
<!-- End footer -->
</footer>
```

Slika 23.: Prikaz koda za formu kontakt (Izvor: Autorski rad)

4.5. Galerija

Korisnicima je dostupna i galerija dosadašnjih radova koja se prikazuje kao animacija otvaranje kartice galerije. Animacije su kreirane u galerija.css dokumentu kao i dizajn polica na kojima „stoje“ slike. Slike su privatno vlasništvo i obrađene su u programu za obradu slika. Na policama su izložene samo istaknute slike,a korisnici imaju mogućnost cijelu galeriju pogledati ukoliko kliknu na „OVDJE“ što ih direktno spaja na Instagram profil salona gdje mogu pronaći sve radove. U nastavku na slikama 25. i 26 možemo vidjeti i kod stranice u kojima se zapravo samo pozivaju definirane funkcije ovisno o tome kako želimo da nam se animacija kreće.

Slika 24.: Prikaz galerije fotografija (Izvor: Autorski rad)

galerija.php galerija.css

```
87
88      <!-- camera -->
89      <div class="roll-camera">
90        <div class="move-camera">
91          <div class="wallpaper"></div>
92            <!-- shelf -->
93            <div class="shelf top">
94              <div class="face top"></div>
95              <div class="face front">
96
97                <a href="#" data-img-url="../photos/1.jpg" class="photocard">
98                  <img src="./slike/nokti2.jpg" alt="">
99                </a>
100               <a href="#" class="photocard">
101                 <img src="./slike/slika4.jpg" alt="">
102               </a>
103               <a href="#" class="photocard">
104                 <img src="./slike/slika9.jpg" alt="">
105               </a>
106
107               <a href="#" class="photocard">
108                 <img src="./slike/slika1.jpg" alt="" style="margin-left:250px">
109               </a>
110
111             </div>
112             <div class="face back"></div>
113             <div class="face left"></div>
114             <div class="face bottom"></div>
115           </div>
116           <!-- /shelf -->
117
118           <!-- shelf -->
119           <div class="shelf middle">
120             <div class="face top"></div>
121             <div class="face front">
```

Slika 25.: Prikaz koda za galeriju fotografija (Izvor: Autorski rad)

Sve klase definirane su u .ccs dokumentu, a navedene slike nalaze se u mapi „slike“. Primjer dijela kreiranja i prikaza polica možemo vidjeti na slici 26.

Pomoću „transform“ pomićemo police u smjeru u kojem želimo. U translate funkciji navodimo jednu ili dvije vrijednosti. Dakle, transformaciju karakterizira dvodimenzionalni vektor. Njegove koordinate definiraju koliko se element kreće u svakom smjeru (4).

Mi smo u svom primjeru koristili jednu vrijednost, a ona nam predstavlja apcisu spoja iz prevođenja vektora.

galerija.php galerija.css

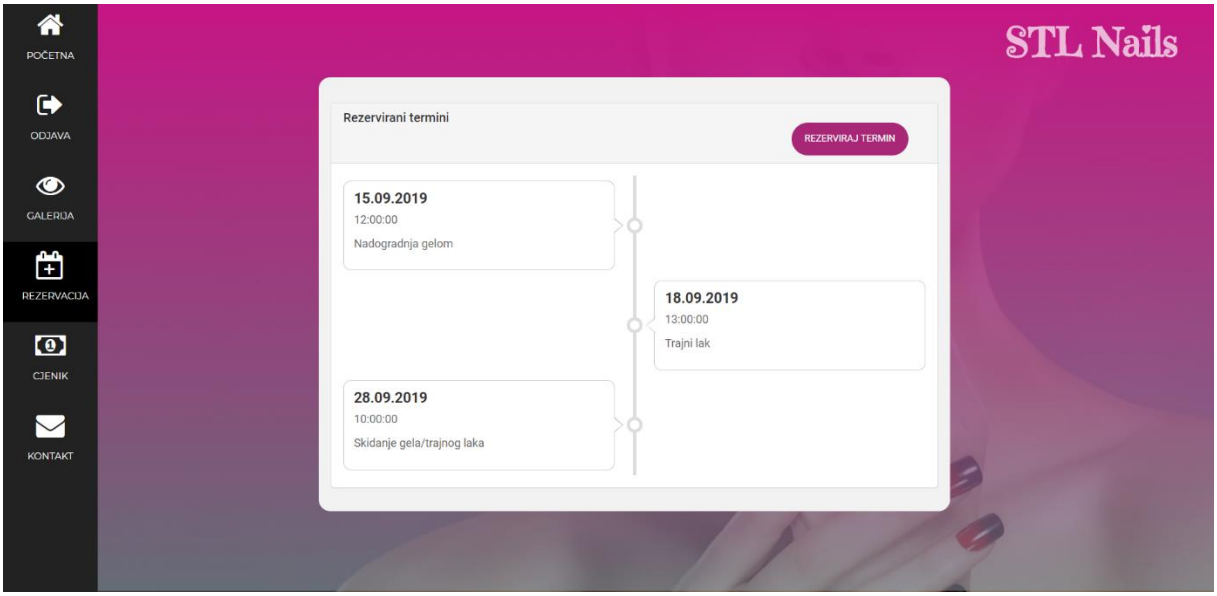
```
191     }
192     .shelf .front {
193         width: 100px;
194         height: 920px;
195         transform-style: preserve-3d;
196         transform: translateX(35px) translateY(-350px) translateZ(10px);
197     }
198     .shelf .back {
199         width: 100px;
200         height: 100px;
201         transform: translateX(35px) translateY(-350px) translateZ(-10px);
202     }
203     .shelf .left {
204         width: 20px;
205         height: 920px;
206         background: linear-gradient(0deg, #eae3d8 0%, #fff7eb 100%);
207         transform: translateX(25px) translateY(-350px) rotateY(-90deg) translateZ(0);
208     }
209     .shelf .top {
210         width: 20px;
211         height: 100px;
212         transform: translateX(75px) translateY(300px) rotateX(90deg) rotateZ(90deg);
213     }
214     .shelf .bottom {
215         width: 20px;
216         height: 100px;
217         transform: translateX(75px) translateY(-400px) rotateX(90deg) rotateZ(90deg);
218     }
219
220     /* lighting */
221     .shelf:before {
222         content:"";
223         display:block;
224         width: 670px;
```

Slika 26.: Prikaz datoteke stila za formu galerija (Izvor: Autorski rad)

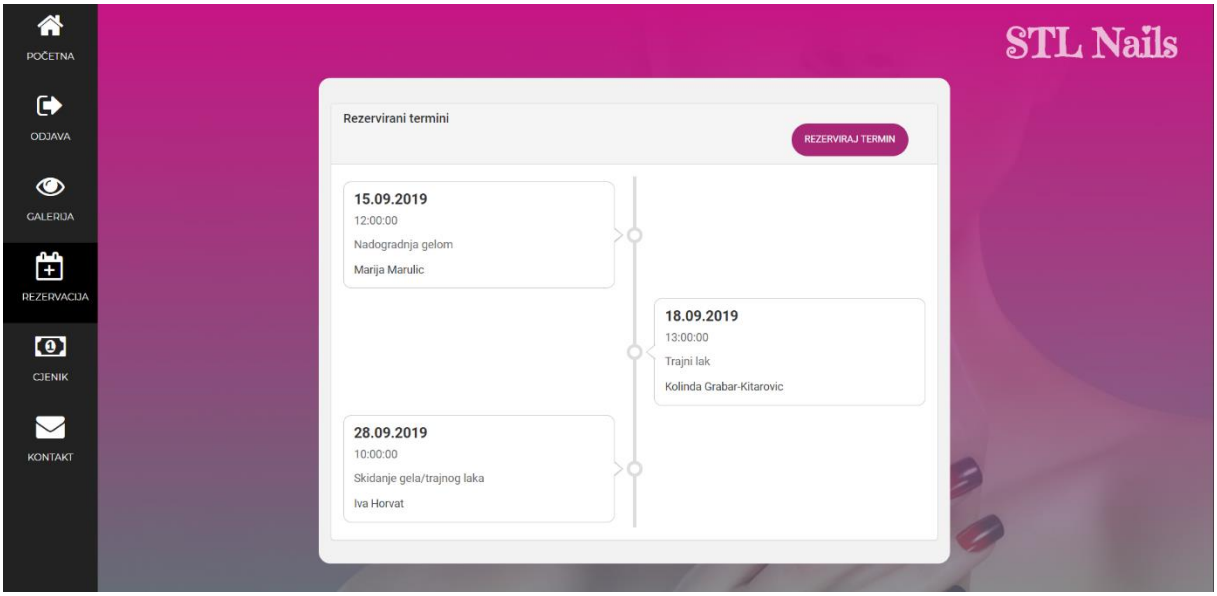
4.6. Rezervacija

Dolazimo i do glavne funkcionalnosti aplikacije, a to je rezervacija termina. Korisnik ima mogućnost pregledati zauzete termine i kreirati svoju rezervaciju termina. Korisniku nije omogućeno vidjeti tko je kreirao rezervaciju već vidi samo u koje vrijeme

je termin zauzet. Na slikama 27. i 28. možemo vidjeti razliku prikaza za admina i za korisnika.



Slika 27.: Prikaz zauzetih termina – korisnik (Izvor: Autorski rad)



Slika 28.: Prikaz zauzetih termina – admin (Izvor: Autorski rad)

Termini su sortirani uzlazno po datumu i vremenu. Pomicanjem stranice prema dolje animirano nam se prikazuju ostali termini koji slijede. Dizajn je kreiran kao vremenska crta termina. Za animiranje koristimo timeline.min.css datoteku u kojoj

definiramo stil vremenske crte.

Na slici 29. prikazan je upit koji se nalazi na vrhu datoteke kojim se spajamo na bazu i iz nje odabiremo potrebne atribute iz tablica. Termini koji su rezervirani prije trenutnog datuma neće biti prikazani. U 18. liniji možemo vidjeti kako su datum i vrijeme sortirani uzlazno.

```
12  <?php
13  $connect = new PDO("mysql:host=localhost;dbname=database", "root", "");
14  $query = "SELECT t.datum, t.vrijeme, u.naziv, k.ime, k.prezime FROM termini t
15            JOIN usluge u ON u.id = t.id_usluge
16            JOIN korisnici k on k.id = t.id_korisnik
17            WHERE t.datum>DATE(now())
18            ORDER BY t.datum, t.vrijeme ASC";
19
```

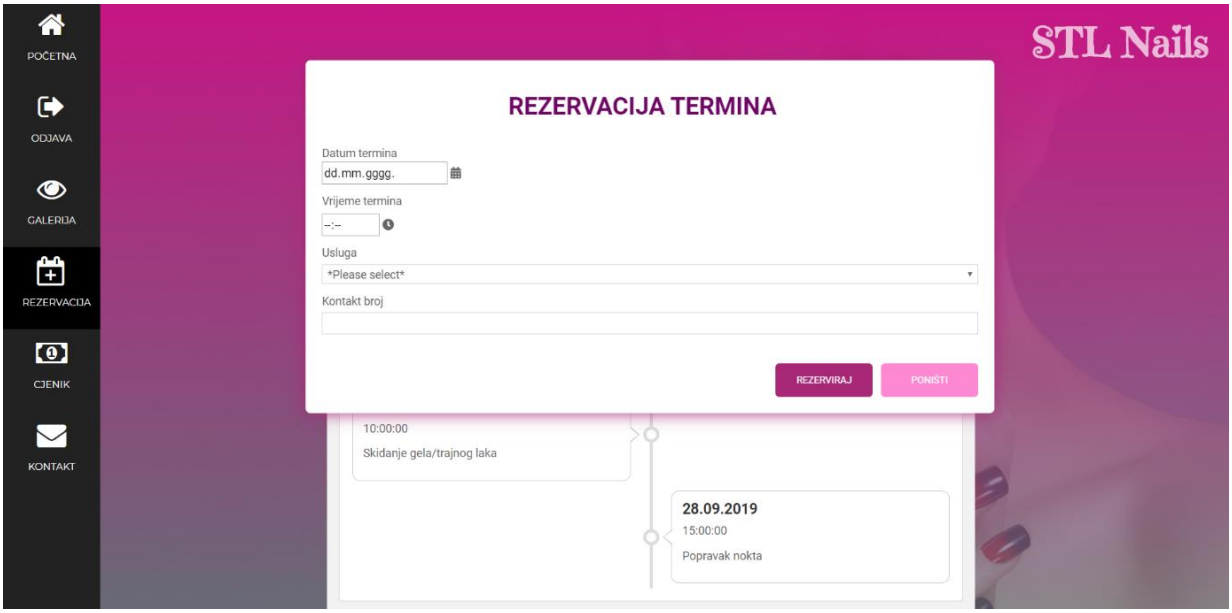
Slika 29.: Prikaz koda za dohvaćanje podataka iz baze (Izvor: Autorski rad)

Na slici 30. prikazan je kod prikaza podataka rezervacija. Od 192. linije koda ispisujemo podatke iz baze. Datumu smo postavili željeni format koji će nam se prikazati na stranici. Vrijeme i naziv usluge se ispisuju nakon datuma, a ime i prezime se ispisuje jedino ako je indeks tipa korisnika jednak broju 3 što bi značilo da je korisnik admin.

```
<div class="container">
  <br />
    <div class="panel panel-default">
    <div class="panel-heading">
        <h3 class="panel-title">Rezervirani termini</h3>
        <div class="btn-block">
            <button type="submit" value="Submit" href="/" onclick="openForm()" style="margin-left
        </div>
    </div>
    <div class="panel-body">
     <div class="timeline">
      <div class="timeline__wrap">
       <div class="timeline__items">
       <?php
       foreach($result as $row)
       {
       ?>
        <div class="timeline__item">
         <div class="timeline__content">
          <h2><?php echo date("d.m.Y", strtotime($row['datum']));?></h2>
          <p><?php echo $row["vrijeme"];?></p>
          <p><?php echo $row["naziv"];?></p>
          <?php
            if($tip == 3){
            echo ($row["ime"]) . "  " . ($row["prezime"]);
            }
          ?>
         </div>
        </div>
       <?php
       }
       ?>
```

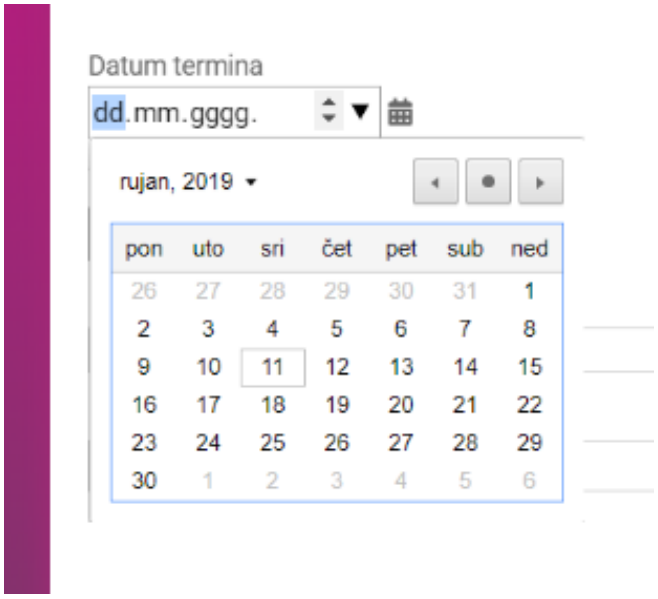
Slika 30.: Prikaz koda za ispis podataka iz baze (Izvor: Autorski rad)

Klikom na gumb „rezerviraj termin“ koji se nalazi u gornjem desnom kutu forme otvara nam se forma za rezervaciju termina koja je prikazana na slici 31.



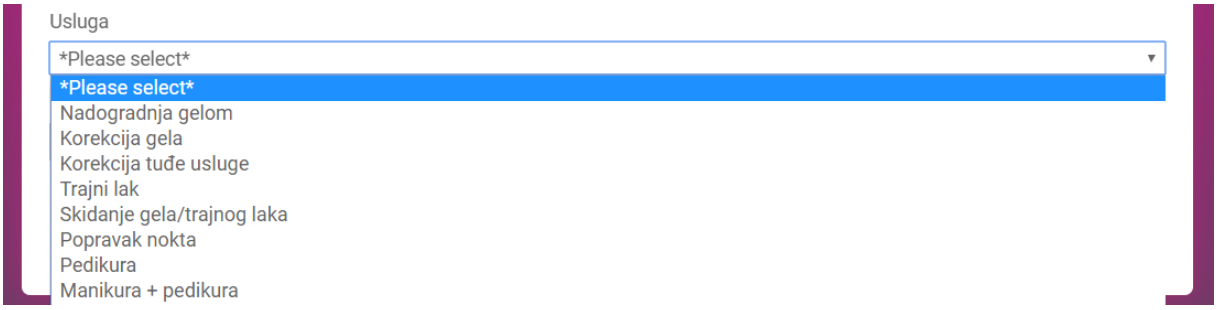
Slika 31.: Prikaz forme za rezervaciju termina (Izvor: Autorski rad)

Forma zahtjeva unos datuma i vremena kada želimo rezervirati termin te izbor usluge i kontakt broj. Datum možemo izabrati i pomoću mini-kalendara kako ne bismo morali upisivati ručno.



Slika 32.: Prikaz okvira za unos datuma (Izvor: Autorski rad)

Također, usluge su nam ponuđene u padajućem izborniku te izbor usluge direktno povlači uslugu iz baze.



Slika 33.:Padajući izbornik usluga (Izvor: Autorski rad)

Prikaz spremanja podataka u bazu vidimo na slici 34.

```php
<?php
include("sesija.class.php");
Sesija::kreirajSesiju();

$datum = filter_input(INPUT_POST, 'datum');
$vrijeme = filter_input(INPUT_POST, 'vrijeme');
$usluga = filter_input(INPUT_POST, 'usluga');
$korisnik = Sesija::dajKorisnika()["id"];
$tip = Sesija::dajKorisnika()["id_tip"];
if (!empty($datum)){
if (!empty($vrijeme)){
if (!empty($usluga)){
$host = "localhost";
$dbusername = "root";
$dbpassword = "";
$dbname = "database";


// Create connection
$conn = new mysqli ($host, $dbusername, $dbpassword, $dbname);
//mysql_query("SET NAMES 'utf8'");
if (mysqli_connect_error()){
die('Connect Error (' . mysqli_connect_errno() .') '
. mysqli_connect_error());
}
else{
$sql = "INSERT INTO termini (datum, vrijeme, id_usluge, id_korisnik)
values ('$datum','$vrijeme', '$usluga', '$korisnik')";
if ($conn->query($sql)){
   header("Location: http://localhost/zavrsni_rad/rezervacija.php");
}
else{
echo "Error: ". $sql ."
". $conn->error;
```

Slika 34.:Prikaz koda za ubacivanje termina u bazu (Izvor: Autorski rad)

Ubacivanje termina u bazu podataka prikazano nam je u 27. liniji gdje smo kreirali SQL upit kojim „insertamo“ vrijednosti datuma, vremena, usluge i korisnika u tablicu termini.

Forma će biti skrivena sve dok je ne pozovemo klikom na gumb „rezerviraj termin“. Klikom na neki od gumba u formi rezervacije termina pozvati će se akcija funkcije za zatvaranje forme koja je definirana na vrhu datoteke i pozivamo je u 251. i 252. liniji koda na klik mišem. Na slici 35. vidimo definiciju funkcije „openForm()“ kojoj smo postavili „display = flex“ koji nam služi za kontrolu izgleda točnije kako je prikazan

element. U našem slučaju pri pozivu funkcije za otvaranje forme će forma ostati fleksibilna, točnije prikazati će se kako smo je definirali, dok će se u funkciji „closeForm()“ forma sakriti zbog svojstva prikaza „none“ koji služi za skrivanje i prikazivanje elemenata bez brisanja i njihovog ponovnog stvaranja (3).

```
57      function openForm() {
58        document.getElementById("myForm").style.display = "flex";
59        //var startD = $.fullCalendar.formatDate(start, "Y-MM-DD");
60        //document.getElementById("dateId").value = startD;
61      }
62
63      function closeForm() {
64        document.getElementById("myForm").style.display = "none";
65      }
66
67      </script>
68    </head>
69
```

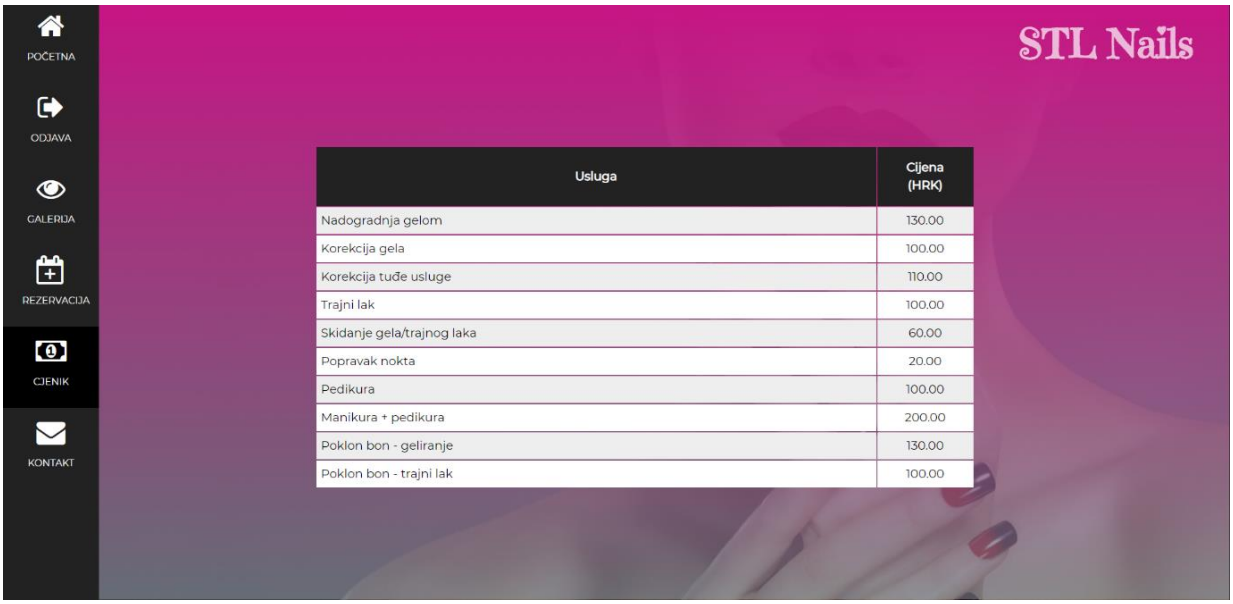
Slika 35.: Prikaz funkcija za otvaranje i zatvaranje forme rezervacije (Izvor: Autorski rad)

```
250         <div class="btn-block">
251           <button type="submit" value="Submit" href="/" onclick="closeForm()" style="display:inline; margin-top: 20px">REZERVIRAJ</button>
252           <button type="reset" value="Reset" href="/" onclick="closeForm()" style="margin-left:0; float:right; background:#fe88d1; margin-
253         </div>
254       </form>
255     </div>
256   </body>
257   </html>
258
```

Slika 36.: Poziv funkcije na klik gumba (Izvor: Autorski rad)

4.7. Cjenik

Korisnik ima uvid u cijene usluga koje se nalaze u tablici klikom na karticu „Cjenik“. Ova kartica je također važna zbog toga što neki korisnici ne znaju kretanje cijena usluga u salonu i ukoliko žele rezervirati svoj termin prvi put onda je dobro da znaju koliko će njihova usluga i koštati.



Slika 37.: Prikaz forme cjenika (Izvor: Autorski rad)

4.8. Odjava

Odjavom korisnika poziva se funkcija obrisiSesiju() koja se poziva na početku odjava.php dokumenta. Pomoću session_unset() i session_destroy() poništavamo sesiju prijavljenog korisnika.

odjava.php

```php
<?php
require_once("sesija.class.php");
Sesija::kreirajSesiju();
Sesija::obrisiSesiju();
header("Location: prijava.php");
die();
?>
```

Slika 38.: Prikaz koda za odjavu (Izvor: Autorski rad)

```php
    static function obrisiSesiju() {
        session_name(self::SESSION_NAME);

        if (session_id() != "") {
            session_unset();
            session_destroy();
        }
    }
```

Slika 39.: Prikaz funkcije za brisanje sesije (Izvor: Autorski rad)

5. BUDUĆI RAD NA WEB APLIKACIJI

Iako potpuno funkcionalna i stabilna, Web aplikacija za upravljanje poslovima kozmetičkog salona ostavlja jako puno prostora za proširenje funkcionalnosti i unapređenje programskog koda, kako optimizacije koda tako i do poboljšanja efikasnosti korisničkog sučelja.

Postoje funkcionalnosti koje nisu implementirane, a u radu aplikacije bi se pokazale korisnima. Kada bi se stavili u ulogu korisnika, dobro bi nam došao dio s recenzijama prethodnih korisnika koji su koristili neku od usluga salona, ocijene noktiju i slično.

Dobra stvar bi bila kada bi korisnici uz povratnu informaciju mogli učitati svoju sliku jer danas kozmetičari uglavnom obrađuju slike kako bi usluga ispala još bolja i kako bi privukli više korisnika. Taj način počeo se koristiti na sve više Web aplikacija koje se bave prodajom proizvoda ili usluga pa jedan takav primjer je „Wish“.

Isto tako bilo bi dobro kada bi korisnici mogli preko aplikacije direktno kontaktirati vlasnika putem nekih chatova, a ne putem e-mail obrasca jer bi to uvelike ubrzalo način komunikacije s obzirom da je u današnje vrijeme takav princip komuniciranja moderniji i više korišteniji.

Pri rezervaciji termina dodali bismo mogućnost promjene termina najkasnije tjedan dana prije i online plaćanje kapare za rezervaciju kako bi se vlasnik osigurao da će se korisnik zbilja pojaviti u vrijeme rezerviranoga termina.

Web aplikacije se zbilja mogu proširiti funkcionalnostima na različite načine. Dobro je pratiti trendove i velik je plus ponuditi korisnicima nešto što ne postoji ili nije često jer ćemo samo tako doći do njihove lojalnosti, a time i većeg profita.

6. ZAKLJUČAK

S obzirom na činjenicu da je razvoj Web aplikacije za upravljanje poslovima kozmetičkog salona tema za izradu završnog rada s naglašenim praktičnim dijelom, odnosno samom implementacijom aplikacije, potrebno je u ovom zaključku osvrnuti se na zahtjeve koji su bili postavljeni spram aplikacije te na razinu do koje su ti zahtjevi ispunjeni. Kao i svaki programski proizvod, Web aplikacija za upravljanje poslovima kozmetičkog salona nije ni približno savršena i njen razvoj u budućnosti će biti nastavljen. Ipak, potrebno je istaknuti da je najveći dio postavljenih zahtjeva ispunjen te su implementirane i funkcionalnosti koje su se pokazale zanimljivim i potencijalno korisnim tijekom samog razvoja.

Najveća prednost tokom izrade aplikacije je istraživanje novih programskih okruženja koja nam omogućuju bolju budućnosti i snalaženje u drugim komplementarnim programskim okruženjima.

Kroz rad je izrađena Web aplikaciju za upravljanje poslovima u uslužnoj djelatnosti, konkretno u radu kozmetičkog salona. S obzirom na kompleksnost pri vođenju rezervacija, javljanja na pozive i ostalo, javila se potreba za smanjenje takvih obveza. Izradom aplikacije stvoreni su uvjeti da korisnici naprave sami taj posao online pomoću iste.

Danas kad uglavnom svi imaju pristup internetu, posebno u urbanim sredinama gdje je i potreba za kozmetičkim uslugama veća, rješenje problema praćenja i narudžbi te većeg dijela administrativnih poslova, mogao bi se riješiti uvođenjem jedne ovakve aplikacije.

Upotrebom PHP tehnologije za pozadinski dio (eng. Backend), te HTML, CSS, Javascript i jQuery tehnologije za vidljivi dio (eng. Frontend) nastojalo se omogućiti korisniku da rezervira svoj termin online, pošalje upit ukoliko ga nešto zanima, pogleda dosadašnje radove i dr.

Kao HTTP poslužitelj odabran je Apache, a za pohranu korištena je MySQL baza podataka. Web Storm kao besplatan alat za uređivanje te oblikovanje projekta i PHP koda pokazao se vrlo praktičnim.

Dizajn Web stranice rađen je kombinacijom HTML koda i CSS stila. Sve slike pohranjene u galeriji su privatno vlasništvo, a obrađene su u programu za obradu slika.

Za modeliranje i oblikovanje baze podataka korišten je XAMPP web server otvorenog koda. XAMPP server se sastoji od MySql baze podataka i Apache web

servera, a ima mogućnost interpretacije skripti u PHP programskom jeziku. Prilikom pokretanja XAMPP-a potrebno je uključenje Apache web servera i MySQL baze podataka kako bi Web aplikacija radila i kako bi njen sadržaj mogao biti promijenjen. Instalacijom XAMPP-a automatski je dobiven i phpMyAdmin koji služi za jednostavno upravljanje bazom podataka.

Izrađena aplikacija je uvelike pomoć pri radu s korisnicima, uspostavljeni su neki novi kanali komunikacije s klijentima. Također, dostupnost iste, što znači, tristo šezdeset pet dana u godini, dvadeset četiri sata dnevno, uvelike povećava konkurentsku prednost poslovnom subjektu te ovaj aspekt ni u kom slučaju nije zanemariv.

LITERATURA

1. Rabuzin, K. (2011.). *Uvod u SQL.* Varaždin: Fakultet organizacije i informatike, Sveučilište Zagreb.
2. Varga, M. (2010.). Modeliranje poslovnih potprocesa i podataka za potporu odvijanja procesa nabave. U *Zbornik radova Međimurskog veleučilišta u Čakovcu* (str. 101-108). Čakovec: Tehnička škola, Čakovec, Hrvatska.
3. W3schools**:** http://w3schools.sinsixx.com/php/default.asp.htm
4. Resources for developers: https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function
5. Coding infinite: https://codinginfinite.com/getting-started-web-application-php-mysql-tutorials/

POPIS SLIKA

Slika 1.: Use case diagram (Izvor: Autorski rad) 5
Slika 2.: ER model (Izvor: Autorski rad) 6
Slika 3.: Početna stranica prije izvršene prijave (Izvor: Autorski rad) 8
Slika 4.: Prikaz početne stranice nakon prijave (Izvor: Autorski rad) 9
Slika 5.: Prikaz koda početne stranice_prvi dio (Izvor: Autorski rad) 10
Slika 6.: Prikaz koda početne stranice_drugi dio (Autorski rad) 11
Slika 7.: Prikaz funkcija unutar sesija.class.php (Izvor: Autorski rad) 13
Slika 8.: Poziv dokumenta sesija.class.php u drugi dokument (Izvor: Autorski rad) . 14
Slika 9.: Forma prijave (Izvor: Autorski rad) 15
Slika 10.: Prikaz poruke praznih polja (Izvor: Autorski rad) 15
Slika 11.: Prikaz poruke blokiranog korisnika (Izvor: Autorski rad) 16
Slika 12.: Prikaz blokiranih korisnika u bazi (Izvor: Autorski rad) 16
Slika 13.: Prikaz provjera i ograničenja uz prijavu (Izvor: Autorski rad) 17
Slika 14.: Prikaz provjera i ograničenja uz prijavu_drugi dio (Izvor: Autorski rad)..... 17
Slika 15. Prikaz forme registracije (Izvor: Autorski rad) 19
Slika 16.: Prikaz poruke krivo unesenih podataka u formi registracije (Izvor: Autorski rad) 19
Slika 17.: Prikaz koda za formu registracije (Izvor: Autorski rad) 20
Slika 18.:Prikaz koda za formu registracije_drugi dio (Izvor: Autorski rad) 21
Slika 19.: Prikaz kontakt forme (Izvor: Autorski rad) 22
Slika 20.: Prikaz forme „Nisam robot“ (Izvor: Autorski rad) 23
Slika 21.: Prikaz poruke netočnog unosa e-mail adrese u kontakt formi (Izvor: Autorski rad) 23
Slika 22.: Prikaz poruke na e-mail adresi vlasnika salona (Izvor: Autorski rad) 24
Slika 23.: Prikaz koda za formu kontakt (Izvor: Autorski rad) 25
Slika 24.: Prikaz galerije fotografija (Izvor: Autorski rad) 26
Slika 25.: Prikaz koda za galeriju fotografija (Izvor: Autorski rad) 27
Slika 26.: Prikaz datoteke stila za formu galerija (Izvor: Autorski rad) 28
Slika 27.: Prikaz zauzetih termina – korisnik (Izvor: Autorski rad) 29
Slika 28.: Prikaz zauzetih termina – admin (Izvor: Autorski rad) 29
Slika 29.: Prikaz koda za dohvaćanje podataka iz baze (Izvor: Autorski rad) 30
Slika 30.: Prikaz koda za ispis podataka iz baze (Izvor: Autorski rad) 31
Slika 31.: Prikaz forme za rezervaciju termina (Izvor: Autorski rad) 31
Slika 32.: Prikaz okvira za unos datuma (Izvor: Autorski rad) 32
Slika 33.:Padajući izbornik usluga (Izvor: Autorski rad) 32
Slika 34.:Prikaz koda za ubacivanje termina u bazu (Izvor: Autorski rad) 33
Slika 35.: Prikaz funkcija za otvaranje i zatvaranje forme rezervacije (Izvor: Autorski rad) 34
Slika 36.: Poziv funkcije na klik gumba (Izvor: Autorski rad) 34
Slika 37.: Prikaz forme cjenika (Izvor: Autorski rad) 35
Slika 38.: Prikaz koda za odjavu (Izvor: Autorski rad) 35
Slika 39.: Prikaz funkcije za brisanje sesije (Izvor: Autorski rad) 35